WHAT IS CLAIMED IS:

1. A method for managing utilization of a unidirectional stack, comprising the steps:

initializing a fixed stack marker and a stack base for said unidirectional stack;

upon fetching a program instruction to be executed in a computing environment, determining if said program instruction involves accessing a location in said unidirectional stack;

if so, further determining whether said location to be accessed is within a valid stack range; and

providing a warning upon determining that said location to be accessed is not within said valid stack range.

2. The method for managing utilization of a unidirectional stack as set forth in claim 1, wherein said computing environment comprises an architectural simulator operable to simulate a target hardware platform.

3. The method for managing utilization of a unidirectional stack as set forth in claim 2, wherein said target hardware platform is selected from the group consisting of a symmetric multiprocessing system, an asymmetric multiprocessing system, a loosely-coupled multiprocessing system, and a tightly-coupled multiprocessing system.

4. The method for managing utilization of a unidirectional stack as set forth in claim 1, wherein said valid stack range is defined by said stack base and a current valid stack pointer associated with said unidirectional stack.

5. The method for managing utilization of a unidirectional stack as set forth in claim 4, wherein said valid stack range includes said valid stack pointer's location.

6. The method for managing utilization of a unidirectional stack as set forth in claim 1, further comprising the step of returning control to a user upon determining that said location to be accessed is not within said valid stack range.

7. The method for managing utilization of a unidirectional stack as set forth in claim 1, wherein said program instruction is operable to perform a read access with respect to said unidirectional stack.

8. The method for managing utilization of a unidirectional stack as set forth in claim 1, wherein said program instruction is operable to perform a write access with respect to said unidirectional stack.

9. The method for managing utilization of a unidirectional stack as set forth in claim 1, further comprising the step of returning control to an interrupt handler upon determining that said location to be accessed is not within said valid stack range.

10. The method for managing utilization of a unidirectional stack as set forth in claim 1, further comprising the step of specifying a direction of growth for said unidirectional stack.

11. A system for managing utilization of a unidirectional stack, comprising:

means to initialize a fixed stack marker and a stack base for said unidirectional stack in a computing environment;

means for determining if a program instruction involves accessing a location in said unidirectional stack, said program instruction being operable to be executed in said computing environment; and

means for providing a warning upon determining that said location to be accessed is not within a valid stack range associated with said unidirectional stack.

12. The system for managing utilization of a unidirectional stack as set forth in claim 11, further comprising means for returning control to a user upon determining that said location to be accessed is not within a valid stack range associated with said unidirectional stack.

13. The system for managing utilization of a unidirectional stack as set forth in claim 11, wherein said valid stack range is defined by said stack base and a current valid stack pointer associated with said unidirectional stack.

14. The system for managing utilization of a unidirectional stack as set forth in claim 13, wherein said valid stack range includes said current valid stack pointer's location.

15. The system for managing utilization of a unidirectional stack as set forth in claim 11, wherein said computing environment comprises an architectural simulator operable to simulate a target hardware platform.

16. The system for managing utilization of a unidirectional stack as set forth in claim 15, wherein said target hardware platform is selected from the group consisting of a symmetric multiprocessing system, an asymmetric multiprocessing system, a loosely-coupled multiprocessing system, and a tightly-coupled multiprocessing system.

17. A method for managing utilization of a unidirectional stack, comprising the steps:

initializing a fixed stack marker, a stack base and a stack pointer for said unidirectional stack;

initializing a high water mark for tracking said stack pointer's location during execution of a program in a computing environment, said high water mark operating to identify said stack pointer's farthest location from said stack base upon completion of said program's execution;

upon fetching a program instruction to be executed in said computing environment, determining if said program instruction is operable to modify said stack pointer's current location to a new location in said unidirectional stack;

if so, further determining whether said new location is within a predetermined stack range; and

providing a warning upon determining that said new location is not within said predetermined stack range.

18. The method for managing utilization of a unidirectional stack as set forth in claim 17, wherein said predetermined stack range comprises a region bounded by said stack base and said stack marker.

19. The method for managing utilization of a unidirectional stack as set forth in claim 17, wherein said predetermined stack range comprises a region bounded by said stack base and said high water mark.

20. The method for managing utilization of a unidirectional stack as set forth in claim 17, wherein said computing environment comprises an architectural simulator operable to simulate a target hardware platform.

21. The method for managing utilization of a unidirectional stack as set forth in claim 20, wherein said target hardware platform is selected from the group consisting of a symmetric multiprocessing system, an asymmetric multiprocessing system, a loosely-coupled multiprocessing system, and a tightly-coupled multiprocessing system.

22. The method for managing utilization of a unidirectional stack as set forth in claim 17, further comprising the step of returning control to a user upon determining that said new location is not within said predetermined stack range.

23. A system for managing utilization of a unidirectional stack, comprising:

means to initialize a fixed stack marker, a stack base and a stack pointer for said unidirectional stack;

means for tracking said stack pointer's location during execution of a program in a computing environment, said means operating to identify said stack pointer's farthest location from said stack base upon completion of said program's execution;

means for determining if a program instruction is operable to modify said stack pointer's current location to a new location in said unidirectional stack; and

means for providing a warning upon determining that said new location is not within a predetermined stack range associated with said unidirectional stack.

24. The system for managing utilization of a unidirectional stack as set forth in claim 23, wherein said predetermined stack range comprises a region bounded by said stack base and said stack marker.

25. The system for managing utilization of a unidirectional stack as set forth in claim 24, wherein said region includes said stack marker's location.

26. The system for managing utilization of a unidirectional stack as set forth in claim 23, wherein said computing environment comprises an architectural simulator operable to simulate a target hardware platform.

27. The system for managing utilization of a unidirectional stack as set forth in claim 26, wherein said target hardware platform is selected from the group consisting of a symmetric multiprocessing system, an asymmetric multiprocessing system, a loosely-coupled multiprocessing system, and a tightly-coupled multiprocessing system.

28. The system for managing utilization of a unidirectional stack as set forth in claim 23, wherein said predetermined stack range comprises a region bounded by said stack base and a high water mark identified by said means for tracking said stack pointer's location.

29. The system for managing utilization of a unidirectional stack as set forth in claim 28, wherein said region includes said high water mark.